

This Page Is Inserted by IFW Operations  
and is not a part of the Official Record

## **BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images,  
please do not report the images to the  
Image Problems Mailbox.**

# PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2000-216689

(43)Date of publication of application : 04.08.2000

(51)Int.Cl. H03M 13/29  
H03M 13/13  
H03M 13/41

(21)Application number : 2000-003003 (71)Applicant : SHARP CORP

(22)Date of filing : 11.01.2000 (72)Inventor : SOMAYAZULU SRINIVASA V

(30)Priority

Priority number : 99 116765 Priority date : 22.01.1999 Priority country : US  
99 363303 28.07.1999

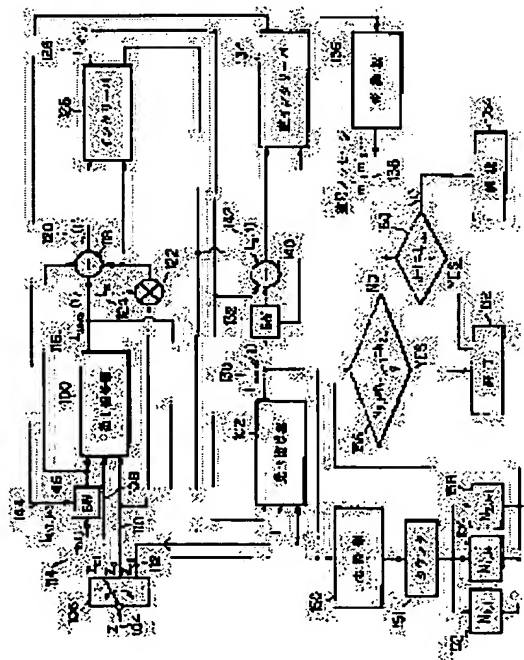
US

## (54) REPETITIVE TURBO CODE DECODER AND METHOD FOR OPTIMIZING PERFORMANCE OF THE DECODER

(57)Abstract:

**PROBLEM TO BE SOLVED:** To provide a turbo code decoder with optimization performance that realizes a prescribed error correction level with a minimum decoding repetitive number in average.

**SOLUTION:** Execution of decoding is monitored by setting up a limit to an output of decoders 100, 102 and measuring number of outputs of one decoder or over nearly equal to this limit. When number of outputs close to the limit or number of saturated outputs is not changed by the successive repetition or successive series decoders 100, 102, further message decoding is not caused. The performance of the turbo code decoder can be optimized by terminating the repetition of decoding process when number of saturated outputs of the decoder is unchanged with respect to the operation of the successive repetition or decoders.



## LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2000-216689

(P 2 0 0 0 - 2 1 6 6 8 9 A)

(43) 公開日 平成12年 8 月 4 日 (2000. 8. 4)

(51) Int. Cl. <sup>7</sup>	識別記号	F I	テーマコード (参考)
H03M 13/29		H03M 13/29	
13/13		13/13	
13/41		13/41	

審査請求 未請求 請求項の数18 O L (全13頁)

(21) 出願番号 特願2000-3003 (P 2000-3003)

(22) 出願日 平成12年 1 月 11 日 (2000. 1. 11)

(31) 優先権主張番号 6 0 / 1 1 6 7 6 5

(32) 優先日 平成11年 1 月 22 日 (1999. 1. 22)

(33) 優先権主張国 米国 (U S)

(31) 優先権主張番号 0 9 / 3 6 3 3 0 3

(32) 優先日 平成11年 7 月 28 日 (1999. 7. 28)

(33) 優先権主張国 米国 (U S)

(71) 出願人 000005049  
シャープ株式会社  
大阪府大阪市阿倍野区長池町22番22号

(72) 発明者 ブイ. スリニバサ ソマヤズル  
アメリカ合衆国, 97223 オレゴン州, デ  
ィガード, 12910 エスタブリュ ウィン  
ターレイク ディーアール.

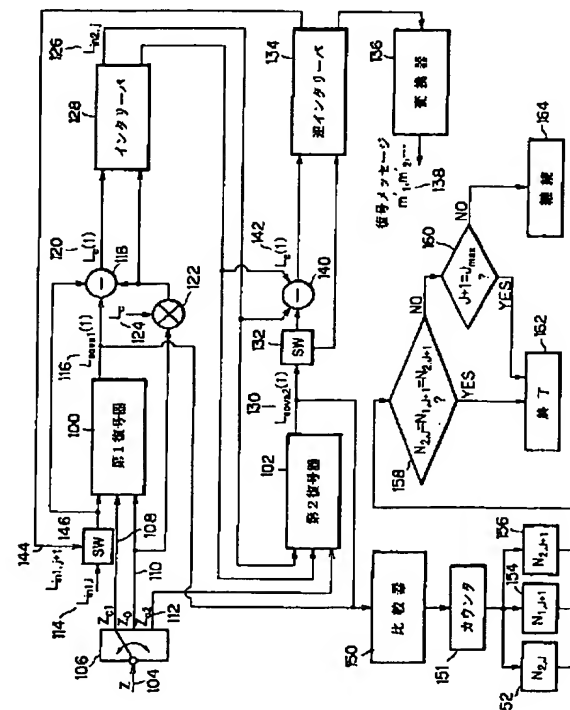
(74) 代理人 100079843  
弁理士 高野 明近 (外 2 名)

(54) 【発明の名称】 反復式ターボ符号復号装置及び該装置の性能を最適化する方法

(57) 【要約】

【課題】 平均して最少の復号反復数で所定の誤り訂正レベルを実現する最適化性能を有するターボ符号復号装置を提供する。

【解決手段】 復号の遂行は、復号器 100、102 の出力に対する制限値を確立し、この制限値にほぼ等しい 1 つ以上の復号器の出力の数を測定することによって監視する。制限値に近似するか又は飽和する出力数が、逐次反復又は逐次直列復号器 100、102 によって変化しなくなれば、メッセージの復号において更なる遂行は無くなる。ターボ符号復号装置の性能は、復号器の飽和した出力数が逐次反復又は復号装置の動作に対し変化しない時に、復号プロセスの反復を終了させることにより、最適化することができる。



## 【特許請求の範囲】

【請求項 1】 複数の復号器を含む反復式ターボ符号復号装置の性能を最適化する方法において、

(a) 前記復号器の出力に対する制限値を確立するステップと、

(b) 前記ターボ符号復号装置による逐次反復の各反復に対して前記制限値にほぼ等しい前記出力の数を決定するステップと、

(c) 前記出力の数が逐次反復間に実質上不変となった時に、前記ターボ符号復号装置の運転を終了させるステップとを含んでなることを特徴とする反復式ターボ符号復号装置の性能を最適化する方法。

【請求項 2】 前記制限値にほぼ等しい前記出力の数は、前記制限値とほぼ等しく該制限値が持つ符号と同じ符号を持つ前記出力の数であることを特徴とする請求項 1 に記載の反復式ターボ符号復号装置の性能を最適化する方法。

【請求項 3】 前記復号器は軟出力ビタビアルゴリズムを用いることを特徴とする請求項 1 に記載の反復式ターボ符号復号装置の性能を最適化する方法。

【請求項 4】 複数の直列復号器を含む反復式ターボ符号復号装置の性能を最適化する方法において、

(a) 前記復号器の出力に対する制限値を確立するステップと、

(b) 第 2 の前記直列復号器が、夫々第 1 回目の反復と第 2 回目の反復を実行中に生成した前記制限値にほぼ等しい前記出力の第 1 の数と第 2 の数を決定するステップと、

(c) 第 1 の前記直列復号器が、前記第 2 回目の反復を実行中に生成した前記制限値にほぼ等しい前記出力の第 3 の数を決定するステップと、

(d) 前記出力の前記第 1 と第 2 と第 3 の数がほぼ等しい時に、前記ターボ符号復号装置の運転を終了させるステップとを含んでなることを特徴とする反復式ターボ符号復号装置の性能を最適化する方法。

【請求項 5】 前記出力の前記第 1 と第 2 と第 3 の数は、前記制限値とほぼ等しく該制限値が持つ符号と同じ符号を持つ前記出力の数であることを特徴とする請求項 4 に記載の反復式ターボ符号復号装置の性能を最適化する方法。

【請求項 6】 前記復号器は軟出力ビタビアルゴリズムを用いることを特徴とする請求項 4 に記載の反復式ターボ符号復号装置の性能を最適化する方法。

【請求項 7】 複数の直列復号器を含む反復式ターボ符号復号装置の性能を最適化する方法において、

(a) 前記復号器の出力に対する制限値を確立するステップと、

(b) 前記復号器が前記反復実行中に生成した実質上全ての出力が前記制限値にほぼ等しい時に、前記ターボ符号復号装置の運転を終了させるステップとを含んでなる

ことを特徴とする反復式ターボ符号復号装置の性能を最適化する方法。

【請求項 8】 前記実質上全ての出力は、前記制限値とほぼ等しく該制限値と同じ符号を持つ前記実質上全ての出力であることを特徴とする請求項 7 に記載の反復式ターボ符号復号装置の性能を最適化する方法。

【請求項 9】 前記復号器は軟出力ビタビアルゴリズムを用いることを特徴とする請求項 7 に記載の反復式ターボ符号復号装置の性能を最適化する方法。

10 【請求項 10】 複数の直列復号器を含む反復式ターボ符号復号装置の性能を最適化する方法において、

(a) 前記復号器の出力に対する制限値を確立するステップと、

(b) 第 1 の前記直列復号器が 1 反復を実行中に生成した前記制限値にほぼ等しい前記出力の第 1 の数を決定するステップと、

(c) 最後の前記直列復号器が前記反復を実行中に生成した前記制限値にほぼ等しい前記出力の第 2 の数を決定するステップと、

20 (d) 前記出力の前記第 1 と第 2 の数が実質上等しい時に、前記ターボ符号復号装置の運転を終了させるステップとを含んでなることを特徴とする反復式ターボ符号復号装置の性能を最適化する方法。

【請求項 11】 前記出力の前記第 1 の数と第 2 の数は、前記制限値にほぼ等しく該制限値が持つ符号と同じ符号を持つ前記出力の数であることを特徴とする請求項 10 に記載の反復式ターボ符号復号装置の性能を最適化する方法。

【請求項 12】 前記復号器は軟出力ビタビアルゴリズムを用いることを特徴とする請求項 10 に記載の反復式ターボ符号復号装置の性能を最適化する方法。

【請求項 13】 反復式ターボ符号復号装置であって、

(a) 複数の直列復号器と、

(b) 該復号器の複数の出力を該出力に対するしきい値と比較する比較器と、

(c) 前記復号器が生成した、前記しきい値にほぼ等しい前記出力の数をカウントするカウンタと、

40 (d) 前記しきい値にほぼ等しい出力の数が逐次反復の間不変である時に、前記ターボ符号復号装置の運転を終了させる判定器とを有する反復式ターボ符号復号装置。

【請求項 14】 前記復号器は軟出力ビタビアルゴリズム復号器であることを特徴とする請求項 13 に記載のターボ符号復号装置。

【請求項 15】 反復式ターボ符号復号装置であって、

(a) 複数の直列復号器と、

(b) 該復号器の出力を該出力に対するしきい値と比較する比較器と、

50 (c) 第 1 の前記直列復号器と後続の前記直列復号器が各々生成した、前記しきい値にほぼ等しい前記出力の第 1 の数と第 2 の数をカウントするカウンタと、

(d) 前記出力の第 1 の数が前記出力の第 2 の数とほぼ等しい時に、前記ターボ符号復号装置の運転を終了させる判定器とを有する反復式ターボ符号復号装置。

【請求項 16】 前記復号器は軟出力ビタビアルゴリズム復号器であることを特徴とする請求項 15 に記載のターボ符号復号装置。

【請求項 17】 反復式ターボ符号復号装置であって、

(a) 複数の直列復号器と、

(b) 該復号器の出力を該出力に対するしきい値と比較する比較器と、

(c) 該しきい値にほぼ等しい前記出力の数をカウントするカウンタと、

(d) 第 1 及び後続の反復を実行する最後の前記直列復号器と、前記後続反復を実行する第 1 の前記直列復号器が各々生成した前記出力で、前記しきい値にほぼ等しい前記出力の第 1 と第 2 と第 3 の数を蓄積するメモリと、

(e) 前記出力の前記第 1 と第 2 と第 3 の数がほぼ等しい時に、前記ターボ符号復号装置の運転を終了させる判定器とを有する反復式ターボ符号復号装置。

【請求項 18】 前記復号器は軟出力ビタビアルゴリズム復号器であることを特徴とする請求項 17 に記載の反復式ターボ符号復号装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明は、反復式ターボ符号復号装置及び該装置の性能を最適化する方法に関し、より詳細には、デジタル通信システム用のチャンネル符号化に有効な反復式ターボ符号復号装置及び該装置の性能を最適化する方法に関する。

【0002】

【従来の技術】 順方向誤り訂正 (FEC) は、伝送チャンネル中のノイズによって引き起こされた“受信状態の”メッセージに含まれている誤りを受信機が検出して訂正できるデータ伝送システム用の誤り制御システムである。FEC は、データの再送を要求できる逆チャンネルを欠くデータ伝送システム、又は遅延が過大なために再送が困難か、或いは、予想される誤り回数のために再送を繰り返す必要があるようなデータ伝送システムに関しては有効である。これらの理由により、FEC は、無線通信及び宇宙探査及び衛星データ通信システムでの使用が特に関心を呼んでいる。FEC は、入力メッセージ系列を写像し、伝送前のデータに冗長性と記憶を付加するシンボル系列を符号化するチャンネル符号化に依拠する。

【0003】 一般に、チャンネル符号化は、ブロック符号化又は通常の符号化を用いてメッセージのビットストリームに冗長性を付加する。ブロック符号化は、メッセージを表現するビットストリームを固定長のブロックに分割し、各ブロックに個別に冗長符号シンボルを付加する。ブロック符号化は通常代数的手法で復号する。他方、畳込み符号化は、ビットストリームにそのコンテン

ツに基づいて連続的に冗長シンボルを付加する。畳込み符号化器において、メッセージのビットは、多数の個別のレジスタを有するシフトレジスタの中に、及びそこから外に順次シフトする。出力符号は、シフトレジスタのコンテンツに対してなされたモジュロ演算の結果であり、幾つかの場合は、入力ビットストリームは各順次メッセージシンボル又はビットとしてレジスタ中でシフトする。ビットストリームのセグメント化は、畳込み符号化の場合必要でないが、代表的には、符号化ビットストリームは伝送前に他の理由からブロック又はフレームに分割される。畳込み符号の復号は、発見的な又は試行錯誤的な方法で達成する。

【0004】 ターボ符号は、2 つ又はそれ以上の並列符号化器より生成される。各符号化器は、しばしば同一の畳込み符号化器であるが、ただし必要条件ではない。符号化器の 1 つ又はそれ以上にインタリーブ又はパーミュータが付加される。インタリーブは、付加符号化器の出力を組織的に擬似ランダムな方式で配置する。結果として、ターボ符号は、符号化器の個数に依存して 2 つ以上の同一入力情報ビットストリームと呼ばれる独立符号化シンボルストリームよりなる。ターボ符号は、比較的単純な成分符号と大きなインタリーブを有し、性能が伝送チャンネルの理論的限界値又はシャノンの限界値に近いので特別な関心を呼んでいる。

【0005】 ターボ符号の復号は、反復プロセスであり、第 1 モジュラ復号器の結果が第 2 モジュラ復号器への入力部分を形成し、以後同様に、所要の反復数に達するまで続いて行く。ターボ符号が 2 つの並列連鎖した符号よりなっている場合、モジュラターボ符号の復号装置は 2 つの直列接続した復号器から構成され、インタリーブが復号器を分離し、第 1 復号器の出力が次の復号器の入力として使用できるように順番を変える。2 つ以上の並列構成符号を有するターボ符号用の復号器は多数の形式を取ることができる。

【0006】 畳込み符号化器は、入力シンボルのシーケンスに基づいて符号ツリー又はトレリス線図を通るパスを追跡することにより符号化する状態機械である。“受信状態”シンボルの符号化メッセージから、畳込み符号の復号器は、伝送中に生じた誤りを訂正しながら復号メッセージのシンボルを出力する符号ツリー又はトレリスを通るパスの再追跡を試みる。畳込み符号を復号する 1 つの技法は、トレリス図を通る“最大尤度”パスを追跡するアルゴリズムに依拠する。ターボ符号の復号に使用する“最大尤度”アルゴリズムの 1 つが軟出力ビタビアルゴリズム (SOVA) である。SOVA アルゴリズムを用いる復号器 (成分復号器) は対数尤比、即ち、観察された信号値を与えられた 2 つの結果 (2 値 “1” 及び “0”) を得る条件付確率比の対数を計算又は推定する。成分復号器の出力は複数の符号の付いた数である。符号は復号メッセージシンボルの極性を表す。大きさ

は、復号シンボルが原メッセージシンボルと同一である確率を表わす“軟”又はアナログ値である。

【0007】

【発明が解決しようとする課題】一般に、ターボ符号の復号装置は、逐次反復により最終復号シンボル系列に収束し、誤り率性能は、反復のしきい値回数に達するまで改善される。復号装置の誤り率性能は、一般に、反復を追加することにより改善されるが、改善率は低下する。各反復は時間を要し復号の完了を遅延させる。これまで、特定のターボ符号復号装置で実行される反復回数は、復号器にハード配線されていた。その復号器にハード配線される反復数の最適化は、復号器の運転時の待ち時間と誤り率の間で妥協点を見つけることを含んでいる。さらに、ノイズのランダムな性質により、“受信状態”のデータ系列は、不同に劣化し、同一レベルの誤り訂正を達成するためには異なる反復数を要する。

【0008】本発明は、上述のごとき実情に鑑みてなされたものであり、平均して最少の復号反復数で所定の誤り訂正レベルを実現する最適化性能を有するターボ符号復号装置及びその復号方法を提供することをその目的とする。また、本発明は、各メッセージの誤り訂正要求によく対応できるよう動作する復号装置を提供することをその目的とする。さらに、本発明は、復号プロセスを最適化することにより、許容誤り率でのメッセージ復号時の待ち時間を短縮し、復号装置のコストと複雑さを軽減させることをその目的とする。

【0009】

【課題を解決するための手段】本発明の第1の技術手段は、複数の復号器を含む反復式ターボ符号復号装置の性能を最適化する方法において、(a) 前記復号器の出力に対する制限値を確立するステップと、(b) 前記ターボ符号復号装置による逐次反復の各反復に対して前記制限値にほぼ等しい前記出力の数を決定するステップと、(c) 前記出力の数が逐次反復間に実質上不変となった時に、前記ターボ符号復号装置の運転を終了させるステップとを含んでなることを特徴としたものである。

【0010】本発明の第2の技術手段は、第1の技術手段において、前記制限値にほぼ等しい前記出力の数は、前記制限値とほぼ等しく該制限値が持つ符号と同じ符号を持つ前記出力の数であることを特徴としたものである。

【0011】本発明の第3の技術手段は、第1の技術手段において、前記復号器は軟出力ビタビアルゴリズムを用いることを特徴としたものである。

【0012】本発明の第4の技術手段は、複数の直列復号器を含む反復式ターボ符号復号装置の性能を最適化する方法において、(a) 前記復号器の出力に対する制限値を確立するステップと、(b) 第2の前記直列復号器が、夫々第1回目の反復と第2回目の反復を実行中に生成した前記制限値にほぼ等しい前記出力の第1の数と第

2の数を決定するステップと、(c) 第1の前記直列復号器が、前記第2回目の反復を実行中に生成した前記制限値にほぼ等しい前記出力の第3の数を決定するステップと、(d) 前記出力の前記第1と第2と第3の数がほぼ等しい時に、前記ターボ符号復号装置の運転を終了させるステップとを含んでなることを特徴としたものである。

【0013】本発明の第5の技術手段は、第4の技術手段において、前記出力の前記第1と第2と第3の数は、前記制限値とほぼ等しく該制限値が持つ符号と同じ符号を持つ前記出力の数であることを特徴としたものである。

【0014】本発明の第6の技術手段は、第4の技術手段において、前記復号器は軟出力ビタビアルゴリズムを用いることを特徴としたものである。

【0015】本発明の第7の技術手段は、複数の直列復号器を含む反復式ターボ符号復号装置の性能を最適化する方法において、(a) 前記復号器の出力に対する制限値を確立するステップと、(b) 前記復号器が前記反復実行中に生成した実質上全ての出力が前記制限値にほぼ等しい時に、前記ターボ符号復号装置の運転を終了させるステップとを含んでなることを特徴としたものである。

【0016】本発明の第8の技術手段は、第7の技術手段において、前記実質上全ての出力は、前記制限値とほぼ等しく該制限値と同じ符号を持つ前記実質上全ての出力であることを特徴としたものである。

【0017】本発明の第9の技術手段は、第7の技術手段において、前記復号器は軟出力ビタビアルゴリズムを用いることを特徴としたものである。

【0018】本発明の第10の技術手段は、複数の直列復号器を含む反復式ターボ符号復号装置の性能を最適化する方法において、(a) 前記復号器の出力に対する制限値を確立するステップと、(b) 第1の前記直列復号器が1反復を実行中に生成した前記制限値にほぼ等しい前記出力の第1の数を決定するステップと、(c) 最後の前記直列復号器が前記反復を実行中に生成した前記制限値にほぼ等しい前記出力の第2の数を決定するステップと、(d) 前記出力の前記第1と第2の数が実質上等しい時に、前記ターボ符号復号装置の運転を終了させるステップとを含んでなることを特徴としたものである。

【0019】本発明の第11の技術手段は、第10の技術手段において、前記出力の前記第1の数と第2の数は、前記制限値にほぼ等しく該制限値が持つ符号と同じ符号を持つ前記出力の数であることを特徴としたものである。

【0020】本発明の第12の技術手段は、第10の技術手段において、前記復号器は軟出力ビタビアルゴリズムを用いることを特徴としたものである。

【0021】本発明の第13の技術手段は、反復式ター

ボ符号復号装置であって、(a) 複数の直列復号器と、  
(b) 該復号器の複数の出力を該出力に対するしきい値と比較する比較器と、(c) 前記復号器が生成した、前記しきい値にほぼ等しい前記出力の数をカウントするカウンタと、(d) 前記しきい値にほぼ等しい出力の数が逐次反復の間不変である時に、前記ターボ符号復号装置の運転を終了させる判定器とを有する反復式ターボ符号復号装置を特徴としたものである。

【0022】本発明の第14の技術手段は、第13の技術手段において、前記復号器は軟出力ビタビアルゴリズム復号器であることを特徴としたものである。

【0023】本発明の第15の技術手段は、反復式ターボ符号復号装置であって、(a) 複数の直列復号器と、  
(b) 該復号器の出力を該出力に対するしきい値と比較する比較器と、(c) 第1の前記直列復号器と後続の前記直列復号器が各々生成した、前記しきい値にほぼ等しい前記出力の第1の数と第2の数をカウントするカウンタと、(d) 前記出力の第1の数が前記出力の第2の数とほぼ等しい時に、前記ターボ符号復号装置の運転を終了させる判定器とを有する反復式ターボ符号復号装置を特徴としたものである。

【0024】本発明の第16の技術手段は、第15の技術手段において、前記復号器は軟出力ビタビアルゴリズム復号器であることを特徴としたものである。

【0025】本発明の第17の技術手段は、反復式ターボ符号復号装置であって、(a) 複数の直列復号器と、  
(b) 該復号器の出力を該出力に対するしきい値と比較する比較器と、(c) 該しきい値にほぼ等しい前記出力の数をカウントするカウンタと、(d) 第1及び後続の反復を実行する最後の前記直列復号器と、前記後続反復を実行する第1の前記直列復号器が各々生成した前記出力で、前記しきい値にほぼ等しい前記出力の第1と第2と第3の数を蓄積するメモリと、(e) 前記出力の前記第1と第2と第3の数がほぼ等しい時に、前記ターボ符号復号装置の運転を終了させる判定器とを有する反復式ターボ符号復号装置を特徴としたものである。

【0026】本発明の第18の技術手段は、第17の技術手段において、前記復号器は軟出力ビタビアルゴリズム復号器であることを特徴としたものである。

【0027】

【発明の実施の形態】本発明は、上述のごとき先行技術の欠点を、複数の復号器を含む反復式ターボ符号復号装置の運転性能を最適化する方法を供給することにより克服する。この方法は、復号器(成分復号器)の出力の制限値を定めるステップと、ターボ復号器による逐次反復ごとの制限値とほぼ等しい復号器出力数を算定するステップと、制限値にほぼ等しい復号器の出力数が逐次反復に対してほぼ不変になった時にターボ符号復号装置の運転を終了させるステップよりなる。メッセージ系列の復号におけるターボ符号復号装置の動作展開は、成分復号

器の出力制限値を決定し、その制限値に等しいか又は近似する出力の数を監視することにより制御できる。もし、制限値に近似又は飽和する出力数がターボ符号復号装置による逐次反復の間変化しなければ、メッセージの復号に進展がないことを示す。したがって、データを失うことなく、ターボ符号復号装置の運転を終了することができる。

【0028】複数の直列復号器を含む反復式ターボ符号復号装置の性能を最適化する第2の技法は、復号器の出力用制限値を確定することと、第1反復及び第2反復を各々実行中に第2直列復号器により生成された制限値にほぼ等しい第1出力数と第2出力数を決定(計算)することと、第2反復中に第1直列復号器により生成された制限値にほぼ等しい第3出力数を決定することと、第1、第2及び第3の出力数値がほぼ等しくなるとターボ符号復号装置の運転を終了させることよりなる。

【0029】複数の直列復号器を含む反復式ターボ符号復号装置の性能を最適化する第3の技法は、復号器の出力用制限値を確定することと、反復を実行する第1直列復号器により生成された制限値とほぼ等しい第1の出力数を決定することと、反復を実行する最後の直列復号器により生成された制限値にほぼ等しい第2の出力数を決定することと、第1と第2の出力数がほぼ等しくなるとターボ符号復号装置の運転を終了することよりなる。

【0030】複数の直列成分復号器を含む反復式ターボ符号復号装置の性能を最適化する第4の技法は、成分復号器の出力用制限値を確定することと、反復を実行中に、直列成分復号器により生成された全ての出力数がほぼ等しくなればターボ符号復号装置の運転を終了することよりなる。復号プロセスの進展を監視し、進展が遅くなるか停止した時に復号の反復を終了することにより、どんなに特殊なメッセージストリームの復号時における遅延も最少にすることができ、同時に、如何なる“受信状態の”シンボル系列についても十分な誤り率を達成できる。

【0031】本発明の最適化反復ターボ符号復号装置は、また、複数の直列復号器と、復号器の複数の出力を出力に対するしきい値と比較する比較器と、しきい値とほぼ等しい復号器の生成出力数をカウントするカウンタと、しきい値にほぼ等しい出力数が、逐次反復に対して不変になった時にターボ符号復号装置の運転を終了させる決定ユニットで構成している。最適化ターボ符号復号装置の第2の実施態様において、反復中に飽和した最初と最後の直列復号器の出力数を比較して復号の進行(遂行)を決定する。第3の実施態様においては、最初の反復を行う最初の直列復号器の出力数を、第1及び後続の反復を行う最後の直列復号器の出力数と比較して、復号の遂行を判断し、さらなる反復が必要か否かを決定する。本発明の上記及び他の目的、特徴及び利点は、以下の詳細な説明を添付の図面を参照して読めば容易に理解



される。

【0032】本発明においては、ディジタル通信システムに使用されるターボ符号を最大尤度アルゴリズムの逐次反復によって復号する。ターボ符号の復号装置は、各モジュールが2台以上の直列復号器よりなるモジュラ装置である。1台の復号器又はモジュールの出力が次の復号器又はモジュールの入力の一部となる。復号の遂行は、復号器の出力に対する制限値を確立し、この制限値にほぼ等しい1台以上の復号器の出力の数を測定することによって監視できる。制限値に近似するか又は飽和する出力数が、逐次反復又は逐次直列復号器によって変化しなくなれば、メッセージの復号において更なる遂行は無くなる。ターボ符号復号装置の性能は、構成復号器の飽和した出力数が逐次反復又は復号装置の動作に対して変化しない時に、復号プロセスの反復を終了させることにより、最適化することができる。

【0033】図1は、チャネル符号化を内蔵する通信リンクを示すブロック図である。前方向誤り訂正(FEC)は、通信リンク2におけるノイズの結果としてディジタルメッセージに発生し得る誤りを検出して訂正するために用いられる。ビット系列又はシンボル系列( $m_1, m_2, \dots, m_j, \dots$ )よりなるメッセージ4は情報源6から生じる。通信リンク2がFECを含んでいれば、メッセージ系列4は、伝送前に符号化装置10により符号シンボル系列8( $C_1, C_2, \dots, C_j, \dots$ )に符号化される。符号語系列8は変調装置12に送られ、通信リンク2のノイズを含む通信の伝送チャネル16で伝送するのに適した信号14( $\{s_j(t)\}$ )に符号化される。伝送チャネル16はノイズの影響を受け受信機20に歪み信号( $\{s_dj(t)\}$ )をもたらす。受信機20において、復調装置22は伝送時に歪みを受けた信号を復調ビットストリーム24( $Z_1, Z_2, \dots, Z_j, \dots$ )に変換する。復調ビットストリーム24は、次に復号装置により復号され、その際、“受信状態の”メッセージ内の誤りが特定、訂正され、訂正され復号化された出力メッセージストリーム28( $m'_1, m'_2, \dots, m'_j, \dots$ )が生成される。出力メッセージストリーム(複合メッセージ)は使用されるために情報シンク30に送られる。

【0034】幾つかのチャネル符号化技法が通信システム内の誤り訂正(FEC)のために使用されている。通信チャネルの理論的な、すなわちシャノンの限界値に近い特性を達成する1つの技法が、ターボ符号化である。ターボ符号は、交錯並行連結符号(interleaved parallel concatenated codes)である。成分符号は畳込み符号であることが多いが、他の技法で生成した符号であってもよい。

【0035】図2は、組織的な反復式畳込み符号化装置の1例を示すブロック図である。片括弧で示したターボ符号符号化装置40の1例は、2つの並列反復式系統的な畳込み符号化器(畳込み成分符号化器)42、44

(各々片括弧で示す)よりなり、追加インタリーブ46が符号化器44の入力に取り付けられている。ターボ符号符号化装置は、2台を超える符号化器で、夫々追加する符号化器の入力にインタリーブを取り付けて構成してよい。このターボ符号符号化装置40は、3つの中間出力である $C_0(47)$ 、 $C_{e1}(48)$ 、 $C_{e2}(50)$ を有し、これらの中間出力は出力多重化器54によって1つの符号化直列ビットストリームに結合される。ターボ符号符号化装置40の符号化率又は入力メッセージシンボル $m$ の数の出力符号シンボル $C$ の数に対する比率は $1/3$ である。

【0036】ターボ符号符号化装置40は、符号化器中間出力の1つ、 $C_0(47)$ を構成する入力メッセージ又はフレーム $m(56)$ のシンボルを有する組織的な符号化装置である。中間出力 $C_{e1}(48)$ 及び $C_{e2}(50)$ は、各々、第1、第2の成分畳込み符号化器42、44の出力である。第1と第2の畳込み符号化器42、44は、各々、1台のシフトレジスタ58と1対のモジュロ2の加算器60、62を含んでいる。シフトレジスタ58は2つの独立したレジスタ $T1(64)$ とレジスタ $T2(66)$ よりなる。符号化器42、44への入力ビットストリームのシンボル $m_j(56)$ 又は $m(\alpha)$ 、(68)は、一度に1ビットずつシフトレジスタ58に送られる。シフトレジスタ58のコンテンツがメッセージの始点において既知であり、またメッセージの全てが完全に各シフトレジスタ58を通して移動するように、ターボ符号符号化装置は、メッセージ又はフレームの終わりの既知状態までシフトレジスタを“フラッシュ”するために十分なビット(通常ゼロ)を加えるパディングユニット70を含む。畳込み符号化器はメッセージに冗長ビットを、メッセージ全体が1つのコードになるような連続条件で加える。しかしながら、符号化メッセージデータは、しばしば、固定長の伝送フレームに分割される。

【0037】ターボ符号符号化装置40において、メッセージ系列 $m(56)$ 、 $m=[10101]$ は第1畳込み符号化器42の入力例である。反復畳込み符号化器42、44において、2つのレジスタ $T1(64)$ 、 $T2(66)$ のコンテンツと入力ビットのモジュロ2の加法計算が、モジュロ2加算器60によりなされ、フィードバックビット $F(72)$ が生成される。畳込み符号化器の出力 $C_{e1}(48)$ 又は $C_{e2}(50)$ は、加算器62で計算された $T2$ レジスタ66のコンテンツとフィードバックビット $F(72)$ のモジュロ2の合計である。第1畳込み符号化器42の出力 $C_{e1}$ は符号化ビットストリーム $[11011]$ である。第2畳込み符号化器44の入力 $m(\alpha)(68)$ は、インタリーブ46によって再配置されたメッセージ系列 $m(56)$ である。インタリーブは、入力シンボルを或る擬似ランダムな方法で再配置し、各入力シンボルを表わす複数のコードシンボルが出



カビットストリームC(52)中で互いに分離されるようにする。誤り訂正を実行するターボ符号は、部分的には、数百ビットをインタリーブする大きなターボインタリーブを使用することに起因する。インタリーブは、同一のメッセージビットストリームに関連する、2つ以上(成分符号数による)の個別に符号化したシンボルストリームを生成する。ターボ符号復号装置は、符号のこの構造を利用し、1つのシンボルストリーム中のビットに関する固有でない(付帯的な)情報を残りのストリーム中のビットから導出し、この情報を用いて、正しく復号されたシンボルを推定する。この推定は復号プロセスの逐次反復の間に洗練される。

【0038】例示のインタリーブ46において、入力シンボルは、入力系列 $m_1$ の第1ビットが畳込み符号化器入力系列68の第4ビット $m(\alpha_4)$ になり、第2入力ビット $m_2$ が第3ビット $m(\alpha_3)$ になるように順次規定したテーブルに基づき再配置される。その結果、第2畳込み符号化器44の出力 $C_{c2}$ (50)、 $C_{c2}=[01100]$ は、符号化器内の符号化プロセスが同じであっても、第1畳込み符号化器42の出力とは異なる。さらに、インタリーブの結果として、入力メッセージシンボルを表わす3つの符号化出力シンボルが出力ビットストリーム52に分離される。

【0039】図3は、図2の符号化装置の反復畳込み符号化器の動作の1例を示すトレリス線図である。畳込み符号化器は状態機械であり、その運転は符号ツリー又はトレリス線図によって追跡されたパスにより表現することができる。トレリス線図の各ノード80は、入力時点又は符号化段階82( $t_0, t_1, \dots$ )における符号化器42のシフトレジスタの状態(トレリスの左の縦列に示した状態)に対応している。2つのパスがトレリスの各ノードを出る。次の符号化段階における次の状態に至るパスは、状態の遷移を惹起するメッセージビットにより決定される。入力メッセージビット“1”は、符号化器に破線で示したパスを辿らせ、入力メッセージビット

“0”は符号化器に実線で示したパスを辿らせる。各パスに対する出力コードシンボル84は、パス付近に示してある。たとえば、 $t_1$ においてビット“1”は符号化器42中でシフトされ、符号化器を初期状態“00”86(パディングの結果)から次の状態“10”88に移行させる。符号化器は、第1ノード“00”86を出発し、次の状態“10”88に行くと、入力“1”に対応するパス(破線)上に出てコードシンボル“11”90を出力する。同様に、次のメッセージビット“0”は、次のコードシンボル“01”92を生成するために第2ノード( $t_1$ における状態“10”)からの出発パスを決定し、符号化器を次の状態( $t_2$ の時点における状態“11”)94に移行させる。符号化器は、全てゼロの状態で起動させ、終了させることが望ましい。入力系列【101】の場合、ゼロ終了状態を実現するために2つ

の余分なビット【01】を必要とする。入力メッセージ系列【101】プラス2つのパディングビット【01】のためのトレリスを通るパスを辿り、出力コード【1101100111】を生成する。図2の符号化器の出力は、出力符号化器(2台の符号化器の各々に対するトレリス線図)の出力であり、原メッセージシンボルと多重化される。

【0040】ターボ符号の復号は、反復プロセスであり、ターボ符号復号装置は、本質的にモジュール構造であり、第1モジュールの出力(第1反復結果)は、次の反復プロセスに用いられる次のモジュールの第1復号器への入力の一部である。

【0041】図4は、本発明によるターボ符号の復号装置を示す図である。例示のモジュール式ターボ符号の復号装置は、特定ターボ符号符号化装置の符号化器(成分符号化器)の数に等しい複数の直列式成分復号器(直列式復号器)100, 102を有している。成分コード数が2つより多い場合は復号器を他の配置も可能であり、本発明はこれらの他の配置構成で使用することも可能である。伝送時に生じた誤りを含んでいる“受信状態”の符号化伝送データZ(104)は、復調装置22から復号装置26で受信される。入力デマルチプレクサ(入力逆多重化器)106において、受信状態データストリームZ(104)は、符号化器の3つの中間出力である $C_0$ (47)、 $C_{c1}$ (48)、 $C_{c2}$ (50)の“受信状態”バージョンを表わす3つの構成ストリームストリームZ。(110)、 $Z_{c1}$ (108)、 $Z_{c2}$ (112)に分離される。

【0042】図5は、軟出力ビタビアルゴリズムに基づいた畳込み復号器の動作の例を示すトレリス線図である。復号器100と102において、軟出力用ビタビアルゴリズム(SOVA)を用いて符号化メッセージを生成した場合にトレリス線図により符号化器が辿った“最尤”パスを再追跡する。図5に示すように、ビタビアルゴリズムを適用する場合、復号器は、プロセスの段階を通して直列作動する。各段階において、パスの距離(メトリック)170を、当初状態からその段階の全ての可能な状態までのパスについて計算する。パスメトリック170は、復号器の当初状態174から特定段階の各ノード176までの各パスに沿った累算差メトリック172である。2つのノード間パスの差メトリック172は、符号化器がこれらの2つのノード又は状態間の遷移をなす際に発生した、受信メッセージのコード語と該当コード語との間の差の測定値である。ビタビアルゴリズムによれば、系列を符号化する際に符号化器が辿ったであろう“最尤”パスは、最小パスメトリックを持つパスである。このパスは次の入力段階において使用するために保持され、メトリックの大きい方のパスは破棄する。このアルゴリズムをさらに深いトレリス線図中の入力段階に適用して行くと、遅延180の後に単一の“最尤”

パスが出現する。受信メッセージ中の誤りが除去され、メッセージがトレリス線図を通るこのパスを辿れば正確に復号できる確率が高い。

【0043】修正ビタビアルゴリズム（軟出力ビタビアルゴリズム、すなわちSOVA）をターボ符号復号装置に用い、軟又は硬な入力から“軟”又はアナログ出力を生成する。“軟”入力又は出力は、可能な信号の二進数値間の実信号レベル（8レベルの場合が多い）を量子化する。“軟”値（軟判定値）は信号値の信頼度とその極性を示している。

【0044】ターボ符号復号装置の復号器（成分復号器）に用いられるSOVAアルゴリズムには、2つの基本的なステップがある。第1ステップ（ビタビステップ）は、トレリスの各段階における各ノードの最大パスメトリック差の計算を追加したビタビアルゴリズムと類似である。第2ステップ（更新ステップ）において、ウィンドウ178を単一パスに収束するビタビのアルゴリズムの遅延に対応して設定する。このウィンドウ178は、復号器が段階から段階に移動するにつれて、トレリス線図に沿ってスライドする。生き残りパスに匹敵し生

$$M_{k,t} = \max_{k'} \left\{ M_{k',t-1} + \frac{1}{2} \sum_{v=1}^N L_c y_{v,t} x_{v,k',k} + \frac{1}{2} L_{in}(t) \right\}$$

【0047】上式において、 $L_{in}(t)$ は、先行復号器から供給される付帯的な情報であり、 $L_c$ は、等式 $L_c = 4a E_s / N_0$ で計算されるチャネルの信頼性ファクタである。式中の $a$ はチャネルのフェージング振幅であり、 $E_s / N_0$ はチャネルの信号対雑音比である。

【0048】図4に示すように、第1復号器100は、SOVAを組織的データ $Z_0(110)$ 、第1復号器100からの符号化データ $Z_{c1}(108)$ 及び初期の先験的な情報 $L_{in1,j}(114)$ に適用し、復号されたメッセージビットの値とその値が正しく復号された確率とを夫々示す複数の符号を付けた数値である軟出力（ $L(sova1)$ ）116を計算する。付帯的情報 $L(e)$ （120）は、減算器118で、復号器の出力 $L(sova1)$ （116）から、乗算器122で組織的データ $Z_0(110)$ にチャネルの信頼性ファクタ $L_c(124)$ を乗じて得た積と先験的な情報 $L_{in1,j}(114)$ を差し引くことにより得られる。インタリーブ後、付帯的情報 $L(e)$ （120）は、先験的な情報 $L_{in2,j}(126)$ になり、第2復号器102に送られる。

【0049】第2復号器102用データは、第1復号器100からのインタリーブして変形した先験情報 $L_{in2,j}(126)$ と、インタリーブ128で同様インタリーブした組織的データ $Z_0(110)$ と、入力デマルチプレクサ106で入力メッセージビットストリーム $Z(104)$ から分離したターボ符号符号化装置の第2符号化器の“受信状態”符号化出力 $Z_{c2}(112)$ とより

き残りパスより異なるビット判定を有する、全ての可能なパスの中から最小のパスメトリック差をウィンドウ内で検出する。最小パスメトリック差には現ビットの硬値に基づく符号が与えられ、SOVA出力の軟値となる。SOVAアルゴリズムを適用する畳込み復号器の出力は、メッセージの各復号ビットに対する符号付き数値よりなる。数値の符号（“対数尤比”の符号）は、ビットの値又は極性を示す。数の大きさ（“対数尤比”の大きさ）は、正しい復号化の信頼度を示す。

10 【0045】SOVAアルゴリズムを用い、ビタビアルゴリズムのパスのメトリック計算は、先行する復号器により供給される付帯的な情報（ $L_{in}(t)$ ）を含む項を追加することにより増大する。トレリス線図中の特定時点 $t$ における各状態 $k$ ごとに、パスメトリック $M_{k,t}$ は、 $t-1$ の時点における全ての各状態 $k'$ から $t$ の時点における状態 $k$ への有効な遷移をもってパスメトリックを延長して下記のように計算する。

【0046】

【数1】

なる。第2成分復号器102は、SOVAアルゴリズムをこの第2符号化情報に適用し、第2軟出力 $L(sova2)$ （130）を生成し、復号プロセスの反復を完了する。

【0050】所要回数の反復を完了すると、第2復号器の軟出力 $L(sova2)$ （130）はスイッチ132により逆インタリーブ134に切り替えられ、コンバータ（変換器）136に送られ、ここで“硬”（二進数）値が、復号装置出力の“軟”（アナログ）値から復号メッセージ138のために選択される。

【0051】所要回数の反復が完了しなければ、第2復号器102の入力からのインタリーブされた組織的データ $Z_0(110)$ とインタリーブされた固有情報 $L_{in2,j}$ を第2復号器の軟出力 $L(sova2)$ （130）から減算器140により減算される。得られた付帯的情報 $L_e(142)$ は、インタリーブで変形させた固有情報 $L_{in1,j,j+1}(144)$ である。逆インタリーブ134で逆インタリーブ後、この固有情報 $L_{in1,j,j+1}(144)$ は、第1復号器100の出力にフィードバックし復号プロセスの次の逐次反復に使用する。

【0052】SOVAアルゴリズムの出力は、生き残りパスに沿う更新ウィンドウ中の最大パスメトリック差中の最小値に依存する。低い信号対雑音比（SNR）の場合、最大パスメトリック差はおそらく小さい。高SNRの場合は、パスメトリック差は、おそらく大きく、SOVA出力値（ $L(sova)$ ）は大きくなる。これらの大き

い値は、アルゴリズムの収束にマイナスの影響を及ぼし、アルゴリズムの性能はこの出力  $L_{sova}$  を制限することにより改善することができる。

【0053】本発明では、SOVAアルゴリズムの収束を、各反復時に制限レベルで飽和し、この数値が反復から反復までの間に変化しなければ反復を中止する復号器の出力数を比較することにより監視できるという予想外の方法を実現した。さらに、単一反復において個別復号器の各々によって生成された飽和出力数の変化を比較することにより、反復を継続するか否かを判定することができる。

【0054】本発明において、出力  $L_{sova}$  は復号器 100, 102 から比較器 150 に送り、ここで、各出力をしきい値又は制限値  $T$  と比較する。しきい値  $T$  と等しくなる、第 1 反復 ( $N2, j$ ) 及び逐次反復 ( $N2, j+1$ ) 中の第 2 復号器 102 からとの出力  $L_{sova}$  の数と、逐次反復 ( $N1, j+1$ ) 中の第 1 復号器 100 の出力  $L_{sova}$  の数は、カウンタ 151 によってカウントし、メモリ 152, 154, 156 に記憶する。3つの飽和値  $L_{sova}$  数を判定器 (判定ユニット) 158 で互いに比較する。3つの数がほぼ等しければ、アルゴリズムは収束したと宣言してよく、ターボ符号復号装置の運転を情報を失うことなく終了できる (162)。3つの数が等しくなれば、アルゴリズムは収束するまで継続し (164)、ターボ符号復号装置も最大反復数に達していない限り (160) 運転を継続する (164)。

【0055】第 2 の技法及び装置は、逐次反復において飽和した第 1 と第 2 の復号器の出力を保管し比較する。飽和出力の数が逐次反復において変化しなければ、収束を宣言し、復号器を停止する。同様に、単一反復器の飽和出力を、飽和出力数に変化が殆ど又は全然無くなるまで、逐次反復中監視することができる。第 4 のより簡単な技法は、最後の直列復号器の出力を最初の直列復号器の出力と比較する。単一反復を行う最初と最後の復号器によって生成した飽和出力の数に変化がなくアルゴリズムの収束が示されるが、この方法を用いた連続運転の効果の確実性はやや劣る。もし、各復号器の SOVA 出力の全ての値が飽和していれば、アルゴリズムは収束し運転を終了できる。しきい値  $T$  と  $L_{sova}$  値は両方とも符号付きの値である。比較回数は、1つの符号を持

つ出力だけを同じ符号を持つしきい値と比較することにより、50%削減できる。

【0056】明細書に用いた用語と表現は、説明のためであり、制限を意味するものではない。かような用語と表現は、特徴又はその部分を表現する同等の用語と表現を除外することを意図したものでは決してない。

【0057】

【発明の効果】本発明によれば、平均して最少の復号反復数で所定の誤り訂正レベルを実現することが可能である。また、本発明によれば、各メッセージの誤り訂正要求によく対応できるよう動作可能である。さらに、本発明によれば、復号プロセスを最適化することにより、許容誤り率でのメッセージ復号時の待ち時間を短縮し、復号装置のコストと複雑さを軽減させることが可能である。

【図面の簡単な説明】

【図 1】チャネルの符号化を内蔵する通信リンクを示すブロック図である。

【図 2】組織的な反復式畳込み符号化装置の 1 例を示すブロック図である。

【図 3】図 2 の符号化装置の反復式畳込み符号化器の動作の 1 例を示すトレリス線図である。

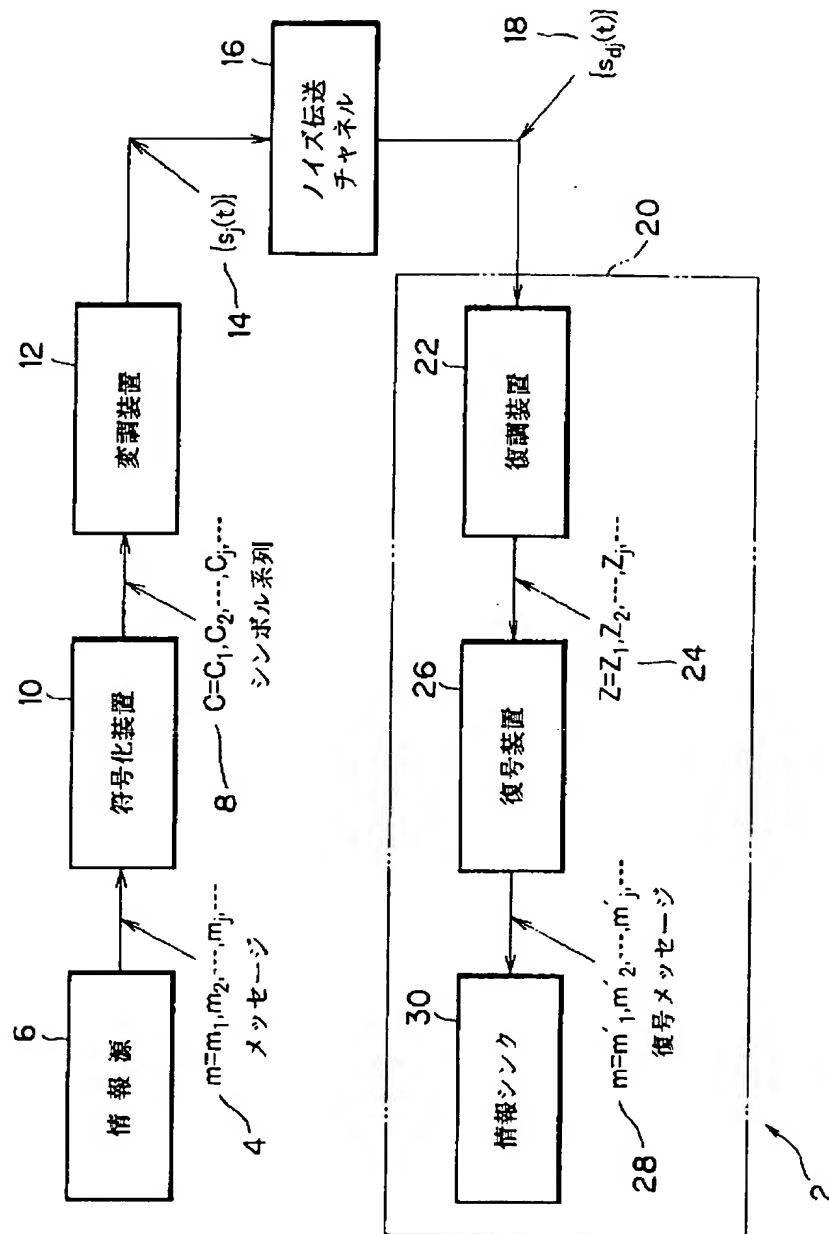
【図 4】本発明によるターボ符号の復号装置を示すブロック図である。

【図 5】軟出力ビタビアルゴリズムに基づいた畳込み復号器の動作の 1 例を示すトレリス線図である。

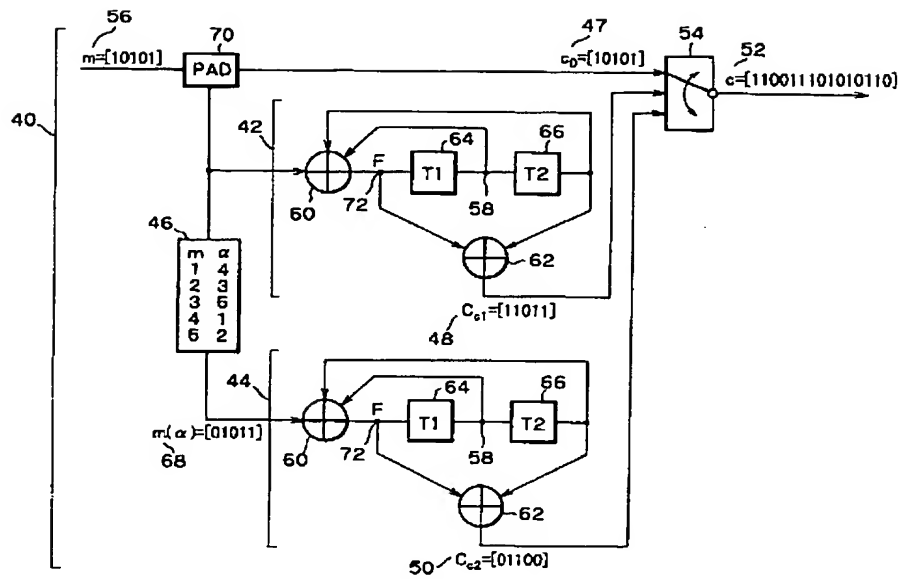
【符号の説明】

4…メッセージ、6…情報源、8…シンボル系列、10…符号化装置、12…変調装置、16…ノイズ伝送チャネル、22…復調装置、26…復号装置、28…復号メッセージ、30…情報シンク、40…ターボ符号符号化装置、42…第 1 符号化器、44…第 2 符号化器、54…出力多重化器、58…シフトレジスタ、60, 62…モジュロ 2 加算器、64, 66…レジスタ、70…パディングユニット、100…第 1 復号器、102…第 2 復号器、128…インタリーバ、134…逆インタリーバ、136…変換器、138…復号メッセージ、150…比較器、151…カウンタ、162 終了、164…継続、170…パスメトリック、172…累算差メトリック、174…当初状態、176…ノード、178…ウィンドウ、180…遅延。

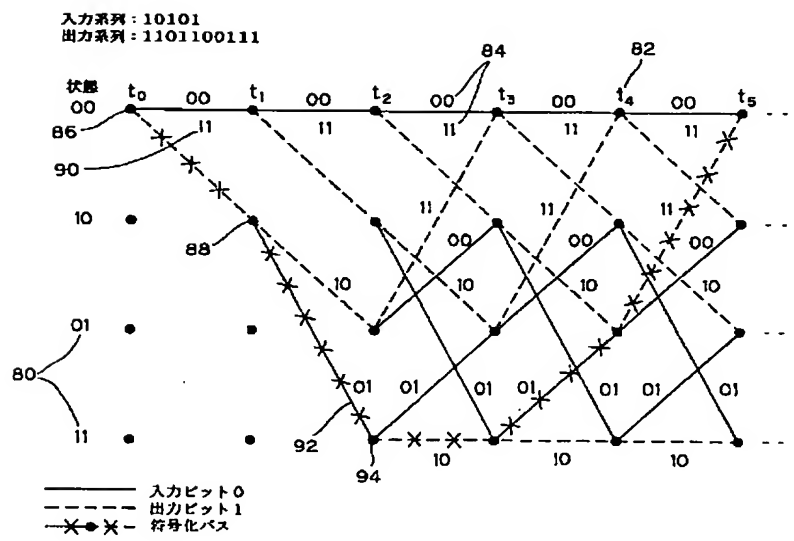
【図 1】



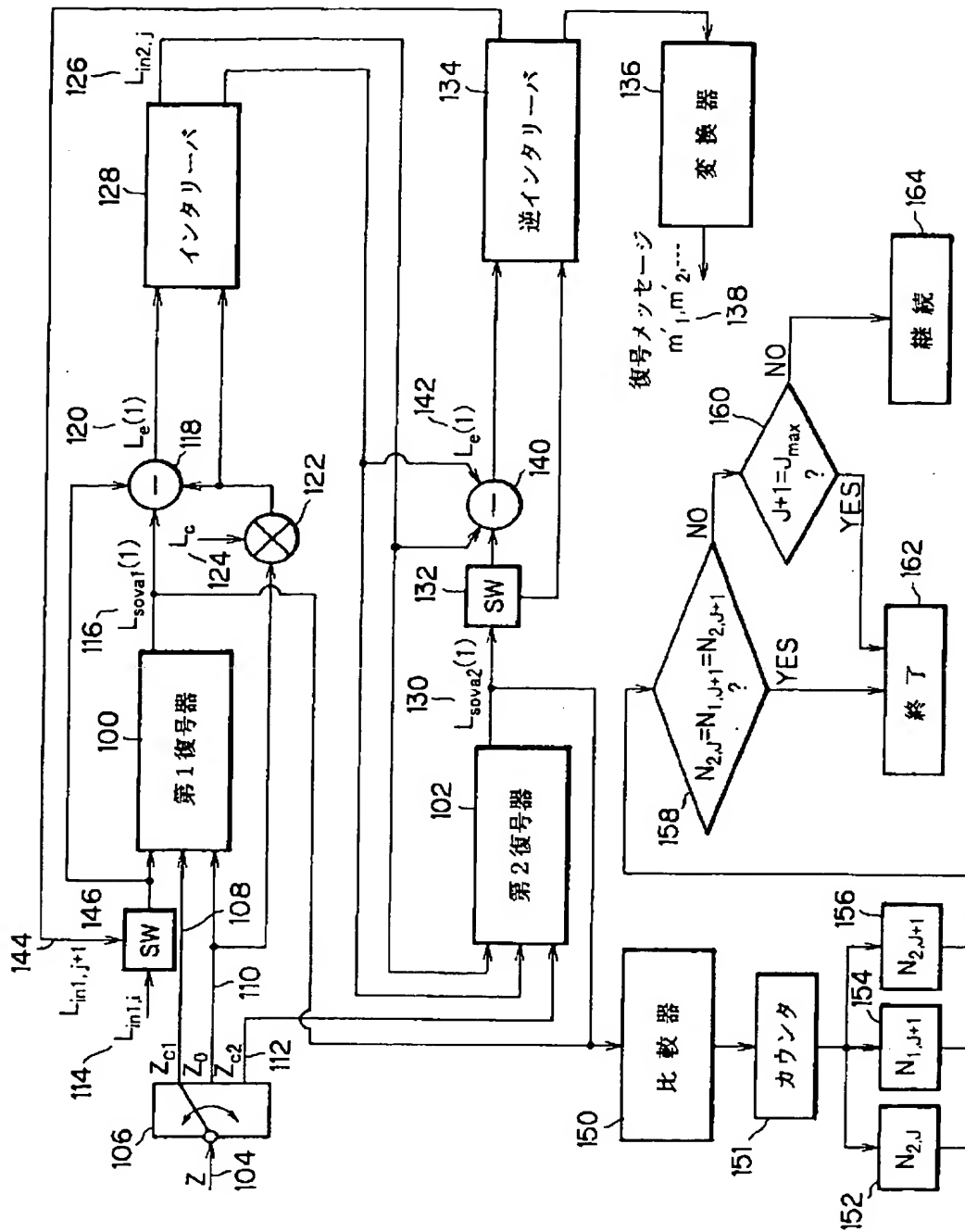
【図 2】



【図 3】



【図 4】



【図 5】

